Ciencia e Ingeniería
Revista Interdisciplinar de Estudios en Ciencias Básicas e Ingenierías.
Año 2015, Enero-Junio, Vol. (2) N° (1) ISSN 2389-9484.
Universidad de La Guajira, Facultades de Ciencias Básicas y Aplicadas e Ingeniería.
La Guajira-Colombia.
Revista en Línea http://revistas.uniguajira.edu.co/index.php/cei



SISTEMA DOMÉSTICO CON INTERFAZ ANDROID USANDO RASPBERRY PI

DOMESTIC SYSTEM INTERFACE WITH RASPBERRY PI USING ANDROID

Eduardo Melchor y Sergio Velásquez*

Centro de Investigación de las Redes Neuronales y la Robótica CIRNAR-UNEXPO
Ciudad Guayana, Bolívar 8050 Venezuela.
sergiovelasquezg@gmail.com

Recibido: Enero 27 de 2015 Aceptado: Junio 03 de 2015

RESUMEN

El presente artículo trata sobre el diseño de un sistema domótico a través de un dispositivo con plataforma Android con conexión a internet, utilizando como módulo de administración y gestión un Raspberry Pi modelo B en enlace con un DSPIC30F4011 mediante el protocolo I2C. De igual forma este se usó como servidor mediante la instalación sobre el sistema Debian del software Apache2 y utilizando como administrador de la base de datos Mysql para lograr el almacenamiento y control de las señales analógicas y digitales en tiempo real a través de los pines GPIO del Raspberry Pi y los diferentes pines del DSPIC30F4011. El manejo de las señales se logró a través de un software diseñado con leguaje de programación python y se realizó la interfaz de la aplicación a través de App Inventor. Se verificó el correcto funcionamiento del módulo creado a través de varias pruebas de monitoreo constante, presencial y de forma remota.

Palabras Clave: Android, Raspberry Pi, Python, App Inventor.

ABSTRACT

This article discusses the design of a home automation system through a device with Android platform with internet connection, using as administration and management module one Raspberry Pi Model B bond with dsPIC30F4011 using the I2C protocol. Likewise this was used as a server by installing on Debian Apache2 system software and using as manager MySQL database to achieve the storage and control of analog and digital signals in real time via the GPIO pin raspberry Pi and various pin dsPIC30F4011. The handling of the signals is achieved through software designed with python programming language of the application interface is performed using App Inventor. Proper operation of the module created through various tests constant attendance monitoring and remotely verified.

Keywords: Android, Raspberry Pi, Python, App Inventor.

1. INTRODUCCIÓN

La domótica es una necesidad de la sociedad moderna para incrementar la calidad de vida de la gente, su aplicabilidad está en reacción directa con la tecnología, ya que para el control y vigilancia de viviendas es necesario que existan dispositivos y medios de comunicación eficientes y confiables para asegurar que la domótica aplicada sea en realidad una solución y no un problema.

En la actualidad existen muchos mecanismos y sensores para el control automático de la vivienda de forma manual, es por ello que el acceso y el control remoto es el siguiente paso para las viviendas inteligentes. De esta forma, el sistema operativo Android como software libre se adapta perfectamente a esta idea, permitiéndonos crear aplicaciones de fácil manejo que se adaptan a nuestras necesidades.

El Raspberry Pi es un ordenador con un sistema operativo de software libre sobre el cual pueden correr varias tareas y que por su tamaño y bajo costo es ideal para ser usado como servidor de aplicaciones domótica, en lugar de utilizar un computador convencional de elevado costo.

Con el presente artículo se profundiza en el estudio del Raspberry Pi, incorporándolo a las posibilidades de diseño como un ordenador de alta tecnología, probando su potencialidad y versatilidad cuando se combina con las nuevas tecnologías como es el caso de Android para el desarrollo de aplicaciones.

Este artículo está estructurado en cinco partes, la introducción para conocer un poco acerca del proyecto, el desarrollo donde se explica cómo se realizó el mismo, los resultados obtenidos, las conclusiones y referencias.

II. DESARROLLO

Propuesta del Diseño.

La propuesta realizada se basa en dos bloques principales de acuerdo a su forma: el hardware conformado por los circuitos integrados y los componentes pasivos; y el software, que contempla la programación en la tarjeta Raspberry Pi y la aplicación en Android.

El sistema está concebido de una manera modular y piramidal, al igual que los sistemas de automatización industrial comúnmente expandidos.

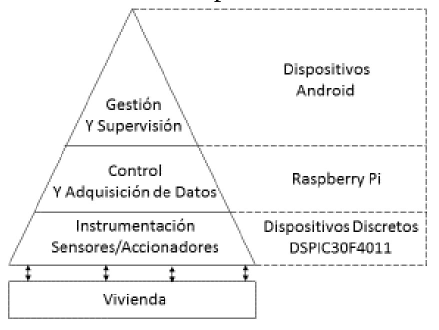


Figura 1: Descripción Conceptual del Sistema

Obsérvese que en la Figura 1, se asume a la vivienda como la planta del sistema por lo cual se hace necesario el diseño de circuitos electrónicos que permitan el acondicionamiento de señales provenir de sensores y circuitos para manejar accionamientos.

La adquisición de datos del sistema requiere de un microcontrolador adicional, debido a la falta de pines de entrada/salida tanto digitales como analógicos en la Raspberry Pi. La etapa de supervisión y control que corresponde a la base de datos para el almacenamiento de eventos o instrucciones, y el control de las entradas y salidas del sistema dependen enteramente del Raspberry PI, mientras que la etapa de gestión que corresponde a la interfaz hombre máquina, y la posibilidad del control remoto del sistema viene dado por la aplicación presente en los dispositivos Android.

Módulo de Gestión y Supervisión.

El módulo de gestión y supervisión se basa en una aplicación desarrollada en plataforma Android de fácil manejo donde se puede monitorear las variables de entrada y controlar las de salida de forma remota en tiempo real conectándose con el módulo de supervisión y control a través del protocolo TCP/IP.

Módulo de Control y Adquisición de Datos.

El módulo de control está constituido por el Raspberry Pi. La aplicación aquí diseñada se comunica con el módulo de entrada/salida mediante el protocolo UART. Figura 2.

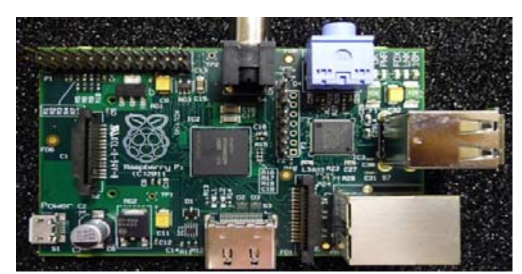


Figura 2: Raspberry Pi Modelo B

El módulo de adquisición de datos está constituido por la Raspberry Pi en enlace con un microcontrolador DSPIC30F4011 dentro del cual se ha instalado una aplicación copilada en lenguaje C. Este microcontrolador por tener una cantidad de pines I/O suficientes para la aplicación requerida y la capacidad de conversión de señales analógicas a digitales, lo hace un dispositivo capaz de cubrir cualquier necesidad prevista para el sistema domótico. Sin embargo, es perfectamente posible seleccionar cualquier otro microcontrolador en función del tamaño de la aplicación domótica.

Módulo de Entrada/Salida.

El módulo de entrada y salida está conformado principalmente por el microcontrolador que cumple la función de accionador inteligente, el cual envía las señales a los

elementos de electrónica discreta o integrada con la finalidad de adecuar las señales físicas a las características de los puertos de comunicación del microprocesador.

Diseño del Hardware.

Este proyecto se separa en dos fases, el "Hardware" que compone la parte física la cual será el medio para recaudar la información que se enviara a la segunda fase y el "Software" que contiene la programación necesaria para procesar y controlar las variables del sistema. Primeramente se diseñara el Hardware, el cual posee cuatro módulos principales que se mostraran a continuación:

- *Módulo de Presencia.*

El elemento principal de este módulo es una fotorresistencia, la cual estará en un divisor de tensión con otra resistencia fija, de tal modo que cuando la iluminación cambie, variaran los valores de voltaje en el divisor de tensión.

Como el propósito de este módulo se basa en la existencia o no de presencia en la habitación, se resume a una salida ON/OFF es por ello que se adaptó a la salida de este divisor de tensión un operacional en modo comparación donde se colocara en el pin positivo un valor referencial que representara la habitación vacía, al variar el valor de la fotorresistencia se entenderá como la intromisión de algún objeto en la habitación, Figura 3.

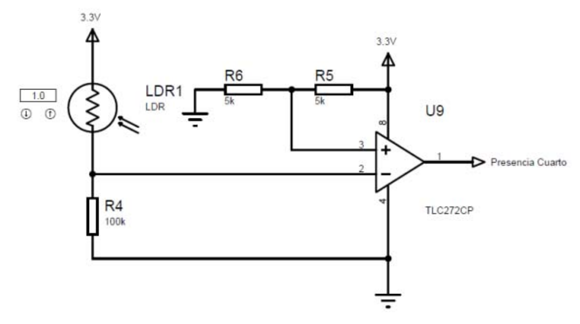


Figura 3: Etapa de acondicionamiento del sensor de presencia

- *Módulo de Portón.*

El objetivo principal de este módulo es el acondicionamiento de un motor DC que se encarga de mover el portón del recinto, este proyecto pretende además, controlar la velocidad de dicho motor, es por esto que se utiliza el módulo PWM presente en la RPi para lograrlo. Sin embargo, para que el motor gire en ambos sentidos es necesario acondicionar la salida del módulo PWM. En la siguiente Figura 4 se detallara como:

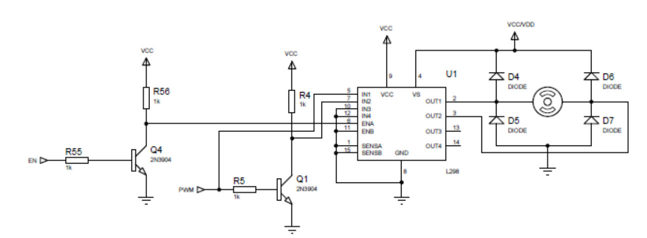


Figura 4: Etapa de acondicionamiento del Motor DC

Por otra parte en la aplicación es necesario saber el estado del portón, si está abierto o cerrado, para ello se utiliza dos opto acopladores, Figura 5.

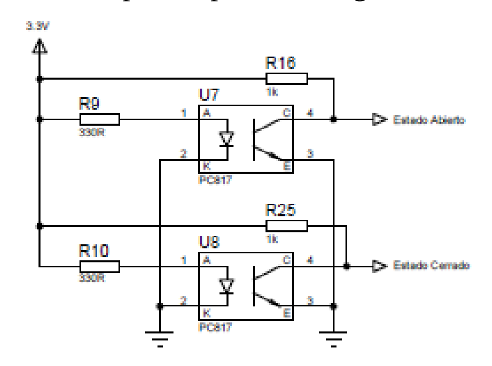


Figura 5: Configuración Básica PC817

- *Módulo de Luces.*

En este módulo se busca acondicionar las señales de entradas tanto locales como remotas para que ambas puedan funcionar en conjunto sin que existan errores de ningún tipo. Estas señales son ordenes de encendido para un bombillo AC, es por ello que también se necesita transformar esa orden de baja potencia en una de mayor escala, para esto se utilizan dispositivos de potencia que nos ayudan a cumplir con la tarea como se muestra en la Figura 6.

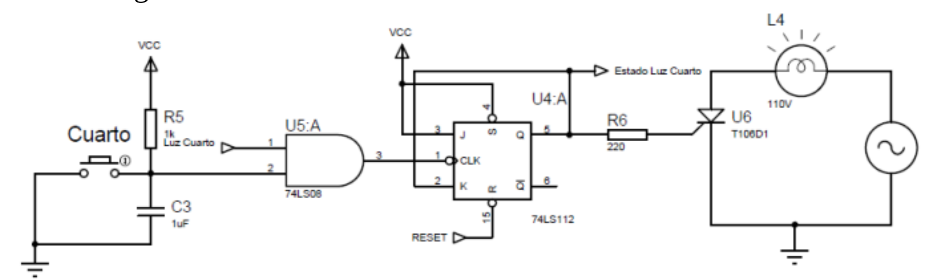


Figura 6: Etapa de acondicionamiento de señales locales y remotas para Luces

- *Módulo de Aires*

Teniendo claro el rango de trabajo para el módulo de temperatura se adecuaron y diseñaron los circuitos para la medida. En este caso el elemento primario es el LM35, este componente ofrece en la salida 10mV por cada grado centígrado de temperatura, de tal forma que a 0 °C -> 0V, y a 50 °C -> 500 mV, por ello antes de llevarla al ADC del DsPIC es necesario amplificar esta señal en un rango de 0V a 5V teniendo una equivalencia de 0°C a 50°C, Figura 8.

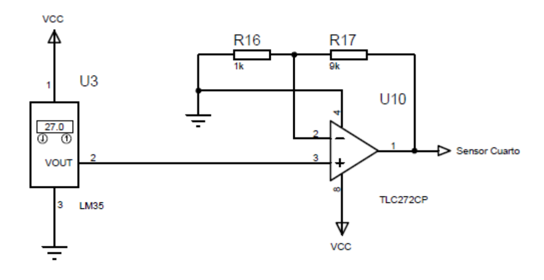


Figura 7: Etapa de amplificación de Salida del LM35

Para el encendido de los aires se utilizó la misma configuración que se usó para el encendido de luces.

Software de Adquisición de Datos.

El software de adquisición de datos consiste fundamentalmente en la comunicación del Raspberry Pi y el DsPIC mediante el protocolo UART, primero se habilitan los pines del puerto serial en la RPi editando el archivo cmdline.txt, borrando los siguientes comandos:

console =ttyAMA0, 115200
kgdboc=ttyAMA0, 115200
Luego se edita el archivo inittab, comentando la siguiente línea:
T0:23:respawn:/sbin/getty –L ttyAMA0 115200 vt100
Finalmente se programan tanto en el RPi como en el DsPIC la misma configuración para el Puerto serial:
Baudios = 9600
Tamaño del byte = 8
Paridad = Ninguna
Bits de Parada = 1

La Raspberry se encarga de recibir todas las entradas mediante los pines GPIO a excepción de las analógicas. Estas son recibidas por el DsPIC que posee un convertidor analógico-digital y luego son enviadas mediante UART al módulo de control. A diferencia de las salidas, las cuales son accionadas por el DsPIC exceptuando la PWM del motor que es controlada directamente por la Raspberry.

Software de almacenamiento de variables.

Para la etapa de almacenamiento se añadió en el sistema una base de datos en MySQL llamada control_hogar, inicialmente en esta base de datos se crearon 3 tablas llamadas Entradas, Salidas y Notificaciones, que permiten comunicación entre los dispositivos.
En la tabla de Entradas se almacena el estado de todos los sensores del recinto, como son: luz, aire, temperatura, presencia y el estado del portón. Con el propósito de facilitar la programación y realizar menos tablas. Algunos campos son compartidos dependiendo de la habitación como se muestra en la Tabla 1

Tabla 1. Datos de Entradas al Software

ID	Zona	Luz	Aire_Abierto	Temperatura_Cerrado	Presencia
1	Habitacion	0	0	0	0
2	Sala	0	0	0	0
3	Cocina	0	0	14.17404	0
4	Garaje	0	0	1	0
5	Externo	0	0	0	0

En la tabla 2, se muestra la tabla de Salidas, la cual se encarga de almacenar las órdenes de accionamientos provenientes de la aplicación en Android para luego ser enviadas al DsPIC para su ejecución.

Tabla 2. Datos de Salidas al Software

ID	Zona	Luz	Aire_Porton	Temperatura_Velocidad
1	Habitacion	false	false	30
2	Sala	false	false	30
3	Cocina	false	false	30
4	Garaje	false	false	5
5	Externo	false	0	0

Por último se tiene la tabla de Notificaciones, la cual almacena el estado de todas las notificaciones de la aplicación con su respectivo correo, al cual se le envía la notificación que esta activada.

Una vez conformada nuestra base de datos, solo queda acceder a ella a través de nuestro código Python en la Raspberry y la aplicación Android.

Para acceder desde el RPi se utilizó una librería capaz de enviar comandos a MySQL directo desde Python, esta librería permite tanto escribir como leer de la base de datos desde el mismo script.

Una vez logrado el acceso a la base de datos desde el Pi, ahora es necesario saber cómo comunicarse con la base de datos a través de la aplicación Android.

Para ello se utilizan tres archivos escritos en lenguaje php: write.php, que nos permite insertar un nuevo correo a la base de datos mediante la función “insert into” de mysql; read.php, este script usa la función “select from” para leer las tablas de la base de datos; y por ultimo update.php, que utiliza la función “update” para actualizar los campos de las tablas. Para acceder desde la aplicación a estos archivos solo hay que colocar: la dirección IP de la Raspberry, el enlace que se desea utilizar y las variables necesarias para llevar a cabo el proceso. Como por ejemplo:

http://192.168.0.106/write.php?Correo=micorreo@gmail.com

Software para el Control y Monitoreo de Variables.

Existen una gran variedad de programas para el desarrollo de aplicaciones Android, sin embargo uno de los más utilizados es App Inventor debido a su programación orientada a objetos, esto hace que la programación sea rápida, fácil y sencilla de aprender. Figura 8.



Figura 8: Interfaz App Inventor

Primeramente en la pestaña "designer" Figura 8, se colocan los botones, campos de texto, imágenes y todo lo que se necesite para controlar la interfaz de la aplicación. Luego, se programa la función de cada elemento en la pestaña "blocks.

Para acceder a la base de datos como se mencionó en el software de adquisición, se coloca un bloque Web y se le asigna a la URL la dirección IP de las Raspberry del enlace que se desea.

Sin embargo, esto solo permite acceder al Pi siempre y cuando el teléfono esté conectado a la misma red. Esto es desfavorable por el hecho de que la finalidad de la aplicación es la de control de forma remota, es decir, el teléfono estará fuera de la red del Raspberry.

Para solucionar este problema, se debe acceder a la dirección pública del router, para que de allí re direccione a la RPi. No obstante, esta dirección varia constantemente, lo que lleva a crear un servidor DNS.

III. RESULTADOS

Funcionalmente el sistema se explica como lo muestra la Fig. 11. Las tablas generadas a través de Mysql contienen tanto el estado de las variables físicas del sistema provenientes de la Raspberry, como las ordenes enviadas por la aplicación en el dispositivo Android. De tal manera que el código en Java comunica la aplicación con Mysql, así como el programa en Python lo hace con el Pi. Figura 9.

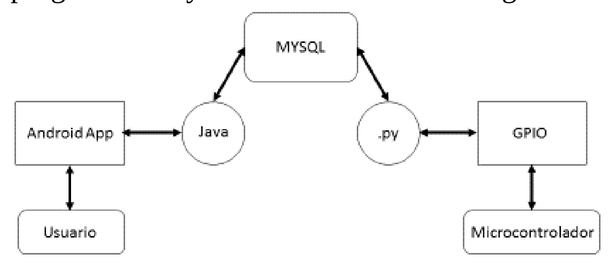


Figura 9: Diagrama Funcional del Sistema

Como resultados tenemos: la aplicación en Android, la maqueta de pruebas y la placa de circuito impreso.

Aplicación Android.

La aplicación posee 4 pantallas diferentes, la pantalla principal, habitación que se repite tanto para sala como para cocina solo que con nombres diferentes, la pantalla del garaje y la de control general del sistema.

En la Figura 10, se muestra la pantalla principal distribuida de la siguiente forma: la habitación en la parte superior derecha, la cocina en la parte superior izquierda, la sala en la parte inferior izquierda y el garaje en la inferior derecha. El color del cuadro indica el rango de temperatura en la cual se encuentra la habitación; azul para menos de 18C, gris para mayores de 18°C pero menos de 22°C y naranja para mayores de 22°C. Los círculos presentes en cada habitación se interpretan como la existencia o no de presencia con el color verde y rojo respectivamente. Los marcos que de momentos están todos en grises es porque no existe luz encendida, en el caso de que lo hubiera, el marco de la habitación donde este encendida cambia a color amarillo. Por último se tiene la barra en la parte inferior del garaje que indica cuando está en rojo que el portón está cerrado y en verde cuando se encuentra abierto.

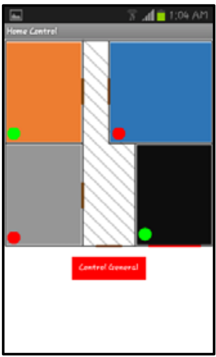


Figura 10: Pantalla Principal

Todas las pantallas presentes en la aplicación tienen la posibilidad de mostrar en la misma el estado de las variables, es decir, si desea encender un bombillo debe saber con anterioridad su estado, a pesar de que este se puede ver en la pantalla principal, cada pantalla tiene la posibilidad de mostrar el estado de la variable antes de cambiarla como se muestra en Figura 11.



Figura 11: Pantalla Control General (a)

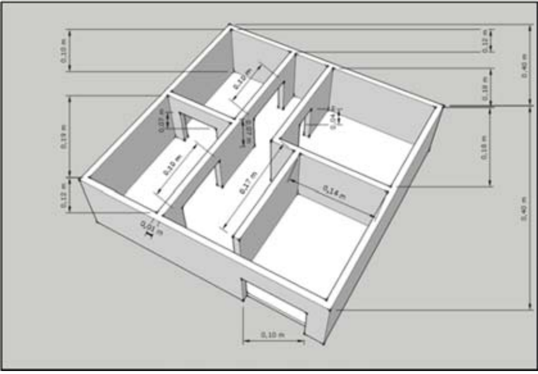
En las Figura. 11 (a) se observa como el botón de las luces Externas e Internas en primer instante están en ON esto es debido a que las luces físicamente están apagadas, si se llegan a encender tanto local como remotamente su estado cambia a OFF Figura 11 (b).



Figura 11: Pantalla Control General (b)

B. *Maqueta de Pruebas.*

Para probar la efectividad de este proyecto se realizó una réplica de una casa estándar con una habitación, una cocina, una sala y un garaje. Cada espacio cuenta con los sensores respectivos descritos en los capítulos anteriores. En la Figura 12, se muestra el plano del recinto diseñado en skectchup.


Figura 12: Plano de Maqueta

Se llevó a cabo la construcción de la réplica con madera y se colocaron los sensores con ayuda de un taladro para abrir agujeros, cables, pinzas y diferentes tipos de pegamentos, Figura 13.



Figura 13: Maqueta de Replica de una Casa Estándar

C. Placa de Circuito Impreso.

Una vez obtenido el circuito final, se realizó la placa de circuito impreso, para ello utilizamos el software que trae Proteus para realizar circuitos impresos, "Ares", en este solo se colocan los componentes en la posición que deseemos, colocamos el tamaño de las pistas y con la opción autoroute, se realizan las pistas de forma automática como se nota en la Figura 14.

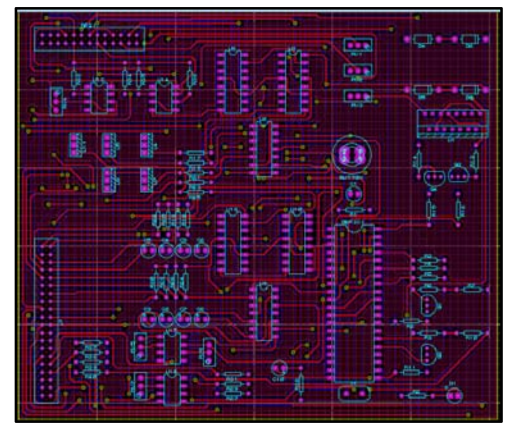


Figura 14: Diseño de Circuito Impreso

Finalmente, tenemos en la Figura 15, la PCB terminada, con dos slots de entrada/salida, uno de 26 pines para el Pi y otro de 40 pines para los sensores y actuadores provenientes de la maqueta.



Figura 15: Placa de Circuito Impreso

Luego, conectamos mediante un cable plano la Raspberry y la maqueta a la placa, así tenemos el proyecto listo para utilizarse en cualquier momento, Fig. 16.

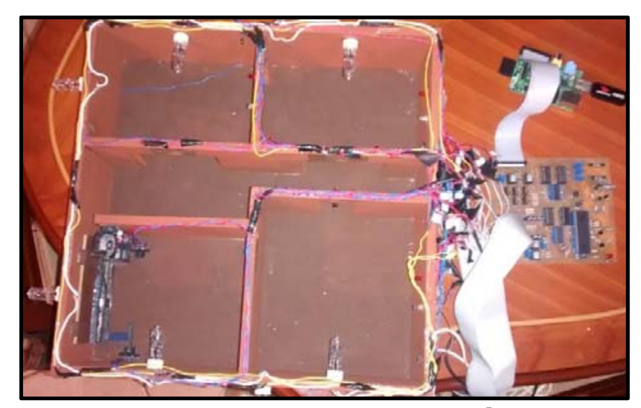


Figura 18: Proyecto terminado

IV. CONCLUSIONES

Tomando en consideración los objetivos del presente trabajo y los resultados obtenidos del mismo se concluye que:
Se estudió el control y monitoreo de variables físicas en el área de domótica a través del módulo Raspberry Pi.
Se estudió la codificación de aplicaciones sobre el sistema operativo Android.
Se diseñó el hardware necesario para el control y monitoreo del sistema.
Se creó y codificó una aplicación sobre el sistema operativo Android de fácil manejo para el control del sistema.
Se implementaron los circuitos diseñados para el control y monitoreo del sistema.
El prototipo sistemático fue implementado con el diseño propuesto en hardware y software, y se realizaron las pruebas necesarias para verificar su desempeño.

V. REFERENCIAS BIBLIOGRAFICAS

ÁLVAREZ, J. M. (2013). Retrieved Agosto 20, 2013, from http://jorgemarcial.com/blog/show/raspberry_pi_tutorial

ALVAREZ, M. (2013). Raspberry Pi ++. Retrieved Septiembre 10, 2013, from rpiplus.blogspot.com

COYÁN. (2012). Retrieved Septiembre 10, 2013, from www.coyan.es/BlogGetting Started with Java SE Embedded on the Raspberry Pi.

GOMEZ, C. (2013). Monitoreo de variables analógicas usando Raspberry Pi. Guayana.

GRAGIRENA,L.(2013). Sistema de supervisión y telemando domótico usando raspberry pi. Guayana.

MICROCHIP. (2007). 2.7V Dual Channel 10-Bit A/D Converter with SPI™ Serial Interface. Retrieved Julio 24, 2013,from http://ww1.microchip.com/downloads/en/DeviceDoc/21294C.pdf